1

Generalized quantifiers as second-order programs — "dynamically" speaking, naturally

ABSTRACT. Generalized quantifiers (understood as functions $\mathcal{Q}$ mapping a set M to a binary relation $\mathcal{Q}_M$ on $\mathrm{Pow}(M)$) are analyzed within the so-called "dynamic" approach to natural language semantics. Under that approach, the meaning of a formula is taken to be the set of input/output transitions it induces, according to a particular translation into programs from some formalism, such as that of (quantified) dynamic logic. The concrete problem addressed here is that variables bound by generalized quantifiers must introduce discourse markers for pronouns (that are typically plural). The solution adopted builds heavily on the "duality" between non-determinism and parallelism by interpreting programs as binary relations on non-empty subsets of states from dynamic logic, thereby accomodating the conjunctive branching in *and-or* computation graphs (which can then be weakened relative to a generalized quantifier). (The distinction between definite and indefinite noun phrases is linked to that between deterministic and non-deterministic programs.) The underlying notion of computation is argued to require a global (as opposed to a local) interpretation, complementing the uniformity in a generalized quantifier. The subsets of interest are hypothesized to have a "determinate" character (i.e., to be first-order definable), in accordance with so-called E-type pronouns. An alternative "indeterminate" reading is also formulated within the framework, as are readings that are weak or strong, and distributive or collective.

INTRODUCTION

An approach to natural language semantics that has attracted, and continues to attract, a good deal of attention is *Discourse Representation Theory* (DRT), initiated in Kamp [**15**] at around the same time (the early eighties) as that of the closely related but independent work of Heim [**13**]. Central to DRT is the proposal of a level of explicit semantic representation called *discourse representation structure*. This feature of the analysis, henceforth referred to as *representational*, was challenged in Barwise [**1**], where the essential point was argued instead to be to

> view the utterance of an expression more dynamically, as having an effect on the environment shared by speaker and hearer, the effect being represented by various sorts of changes in variable assignments. (p. 2)

Shortly afterwards, Groenendijk and Stokhof [**11**], confining their attention to first-order formulas, defined a simple non-representational translation, dubbed *Dynamic Predicate Logic* (DPL), of first-order formulas into programs from

(Quantified) Dynamic Logic (e.g., Harel [**12**]). More recently, further developments of DRT have appeared in the textbook Kamp and Reyle [**14**], far outstripping Kamp [**15**], Barwise [**1**] and Groenendijk and Stokhof [**11**]. These include an account of generalized quantifiers and plurals (going back to 1985/86) reported in chapter 4 of the book. Although certain notions in that chapter admit straightforward transcriptions to a DPL-framework (e.g., the two different semantics of *duplex conditions* in Kamp and Reyle [**14**], that are christened *weak* and *strong* readings in Chierchia [**5**]) and although some interesting extensions of DPL are suggested in Van den Berg [**2, 4**] (as well as Chierchia [**5**] and Van Eijck and De Vries [**7**]), it is not clear that a non-representational treatment of generalized quantifiers comparable in scope and perspicuity to that in Kamp and Reyle [**14**] has been worked out. The present paper is intended as a contribution towards that end.

The linguist is, of course, entitled to ask *why bother* developing yet another formalism with the same empirical coverage? In fact, it is not obvious to the present author that the scope of the proposal below is confined to chapter 4 of Kamp and Reyle [**14**]. But at any rate, the hope is that different formal approaches reflect different intuitions that throw light on issues with genuine "linguistic" content. An intuition stressed by the "dynamic" approach (exemplified by DPL) is that the meaning of a statement lies not so much in its *truth* relative to a context, but in the *change* in context that it induces. Formalizing statements as formulas, and contexts as states, the idea is captured by the slogan "formulas-as-programs", with a program π reduced semantically to its binary input/output relation $\rho(\pi) \subseteq S \times S$ on some set S of states. As will become clear below, the same scheme can be applied to Kamp [**15**], with the essential exception that the relevant states in Kamp [**15**] are the discourse representation structures mentioned above.

By contrast, to support a translation from first-order formulas to programs in quantified dynamic logic, the states in DPL are defined, relative to a first-order model $\mathbf{M}$, to be functions, henceforth called ($\mathbf{M}$-)*substitutions*, from a set of variables to the universe of $\mathbf{M}$. A substitution is nothing more than the "variable assignment" mentioned in the quotation above from Barwise [**1**], and is also referred to elsewhere as an "environment" or a "valuation." Although our choice of the term "substitution" is largely arbitrary, it can be motivated in part by (i) its compatibility with S, (ii) the unmistakable distinction made from certain programs (to be described shortly) called "random assignments", and (iii) the suggestion (e..g, from Barwise [**1**]) that an object in S have as its domain a finite subset of the full (infinite) set X of variables. (This suggestion departs form the conventions of dynamic logic, which are otherwise observed below.) While the present paper will refrain from making a commitment to (iii), except at one point indicated below, let us note in passing that item (iii) yields a more computationally accessible (i.e., realistic) picture that is also more faithful to DRT (in contrast, that is, to the usual convention that a variable assignment is defined on all of X). Keeping (iii) in mind, let us characterize the interpretation of a *random assignment* $x :=?$, where $x \in X$, by the condition that for all f and $g \in S$,

$$\begin{array}{lll} f\ \rho(x :=?)\ g & \text{iff} & x \in dom(g) \text{ and } f = g \text{ except} \\ & & \quad \text{possibly on } x \\ & (\text{iff} & dom(g) = dom(f) \cup \{x\} \text{ and} \\ & & (\forall x' \in dom(f) - \{x\})\ f(x') = g(x'))\ . \end{array}$$

All changes to a substitution in DPL can be traced to such a random assignment, which is used to interpret existential quantification $\exists x$. Conjunction & and negation $\sim$ are analyzed in terms of sequential composition (written ;) and a test for divergence (written $\neg$, instead of the somewhat more cumbersome notation $([\cdot]\bot)?$ of dynamic logic), respectively. The remaining logical connectives (including the universal quantifier) can then be derived as abbreviations in the familiar manner, sound for classical logic.

The details will be presented more carefully below, but for now let us stress the computational and dynamic aspects of DPL: a formula φ is translated into a program φ^{DPL}, resulting in an interpretation $[\varphi]$ of φ as the input/output relation $\rho(\varphi^{DPL}) \subseteq S \times S$. DPL's interpretation of negation, however, features two disturbing conjuncts,

$$f[\sim\varphi]g \quad \text{iff} \quad f = g \text{ and } f \notin dom[\varphi]\ ,$$

belying the propaganda above about change and computation, and raising two somewhat embarrassing questions.

(Q1) How well is the view that "meaning lies in change" served by reducing

$$\varphi \supset \psi \quad \text{to} \quad \sim(\varphi \& \sim\psi)$$

and

$$\forall x\varphi \quad \text{to} \quad \sim\exists x \sim\varphi\ ,$$

given that the negation of a program can only output its input

$$f[\sim\varphi]g \quad \text{implies} \quad f = g$$

(assuming it halts at all)? Do not such programs trivialize the shift from a static (truth-centered) semantics to a dynamic semantics?

(Q2) How seriously can the term "program" be taken, given that (under a recursion-theoretic conception) testing for divergence (i.e., the complement of the halting problem),

$$f \notin dom[\varphi]\ ,$$

falls beyond the realm of computation? In what sense are such programs computational?

We will address points (Q1) and (Q2) somewhat indirectly, by first focussing on a crucial feature of some programs in DPL — non-determinism.

A program π is *non-deterministic* if its input/output relation is *not* functional — i.e., if on some input, π can return one of at least two distinct outputs. Such a possibility is hard-wired in dynamic logic by the binary program connective $+$, which is interpreted as union

$$f\ \rho(\pi + \pi')\ g \quad \text{iff} \quad f\ \rho(\pi)\ g \ \text{ or } \ f\ \rho(\pi')\ g\ .$$

In Peleg [**16**], a binary program construct $\cap$ dual to $+$ is introduced yielding a notion of parallelism. Roughly put, if π and π' are programs from ordinary dynamic logic, then

$$f\ \rho(\pi)\ g \ \text{ and } \ f\ \rho(\pi')\ h \quad \text{imply} \quad f\ \rho(\pi \cap \pi')\ \{g, h\}\ ,$$

the intuition being that whereas $+$ is disjunctive, $\cap$ is conjunctive (which should be kept in mind when specifying transitions from $\{g, h\}$). More rigorously, $\cap$ is accomodated by lifting the function ρ to a function $\hat{\rho}$ with $\hat{\rho}(\pi) \subseteq S \times \mathrm{Pow}_{+}(S)$, where $\mathrm{Pow}_{+}(S) = \mathrm{Power}(S) - \{\emptyset\}$. In fact, we will not have any use for $\cap$, which (for the record) is governed by

$$\begin{aligned} f\ \hat{\rho}(\pi \cap \pi')\ U \quad \text{iff} \quad & (\exists V, W)\ U = V \cup W \ \text{ and} \\ & f\ \hat{\rho}(\pi)\ V \ \text{ and } \ f\ \hat{\rho}(\pi')\ W\ . \end{aligned}$$

Nonetheless, the passage from ρ to $\hat{\rho}$ (along the embedding $s \mapsto \{s\}$ of S into $\mathrm{Pow}_{+}(S)$) will be crucial to the constructs we will introduce, as will Peleg [**16**]'s clause for sequential composition

$$\begin{aligned} f\ \hat{\rho}(\pi; \pi')\ U \quad \text{iff} \quad & (\exists V)\ f\ \hat{\rho}(\pi)\ V \ \text{ and there is a function} \\ & F \text{ with domain } V \text{ s.t.} \\ & \qquad (\forall g \in V)\ g\ \hat{\rho}(\pi')\ F(g) \ \text{ and} \\ & \qquad U = \bigcup \{F(g) \mid g \in V\}\ . \end{aligned}$$

To understand this clause, think of a substitution as a process (or some snapshot of a process), and regard a set of substitutions (such as U and V above) as a family of proceses running in parallel. The universal quantification $(\forall g \in V)$ above formalizes the conjunctive nature of parallelism.

Our move from S to $\mathrm{Pow}_{+}(S)$ will be necessitated by the non-determinism not in $+$, but in random assignments $x :=?$. To a first approximation, the idea is to introduce a unary program construct $\cdot_{\forall}$ such that on programs π from dynamic logic,

$$f\ \hat{\rho}(\pi_{\forall})\ U \quad \text{iff} \quad U = \{g \in S \mid f\ \rho(\pi)\ g\} \neq \emptyset\ ,$$

and, in particular,

$$\begin{aligned} f\ \hat{\rho}((x :=?)_{\forall})\ \{g \in S \mid x \in \mathit{dom}(g) \text{ and } f = g \\ \text{except possibly on } x\}\ . \end{aligned}$$

(The effect of $\cdot_{\forall}$ can be pictured as turning the disjunctive branching of a non-deterministic program to the conjunctive branching of a deterministic program.) As $\hat{\rho}$ is somewhat uneven in that the left and right arguments of

$\hat{\rho}(\pi)$ have different types, it is natural to lift $\hat{\rho}$ further to a map $[\![\cdot]\!]$ that interprets a program π as a binary relation $[\![\pi]\!] \subseteq \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S)$. (The restriction to non-empty subsets of S is made because $\emptyset$ arises from the impossibility of making a ρ-transition.) Just how the lifting is carried out will require some elaboration, but very briefly, the point is that π becomes a second-order program that relates sets of substitutions operating in the conjunctive manner alluded to above. The unary program construct $\cdot_\forall$ can be derived from a binary program construct $\Rightarrow$ introduced in Fernando [**10**] to extend the DPL translation by the clauses

$$\begin{aligned}(\varphi \supset \psi)^{DPL} &= \varphi^{DPL} \Rightarrow \psi^{DPL} \\ (\forall x \varphi)^{DPL} &= (x :=?) \Rightarrow \varphi^{DPL} .\end{aligned}$$

This extension to DPL is "conservative" in that

$$\begin{aligned}\sim\sim(\varphi \supset \psi) &\equiv \sim(\varphi \& \sim\psi) \\ \sim\sim(\forall x \varphi) &\equiv \sim\exists x \sim\varphi ,\end{aligned}$$

where $\varphi \equiv \psi$ means that φ and ψ are interpreted by the same binary relation (i.e., $[\varphi] = [\psi]$). Note that the double negation $\neg\neg\pi$ of a program π is interpreted as $\{(f, f) \mid f \in dom[\![\pi]\!]\}$. The troublesome negation $\neg\pi$ of a program π can be recast as $\pi \Rightarrow \bot$, where $[\![\bot]\!] = \emptyset$. But, in general, the point is that programs of the form $\pi \Rightarrow \pi'$ externalize the dynamic effects of programs $\neg(\pi; \neg\pi')$ used in DPL to interpret implication. In short, the construct $\Rightarrow$ forms our reply to (Q1).

As for (Q2), let us first try to put to rest any doubt about its significance. Referring to dynamic approaches to generalized quantifiers, Van den Berg [**3**] asserts that Van den Berg [**4**] "shares with Chierchia and Van Eijck" ([**5, 7**]) the property "that the dynamic behavior of the quantifier is totally **ad hoc**. The definition gives a condition in terms of the generalized quantifier and on top of that it gives some condition on the output ... There is no reason why either of these definitions, or any of the other proposed in the literature, is natural, let alone the correct one" (p. 2). Such reservations suggest that the computational intuitions behind the term "program" might be worth developing. In fact, the idea of using sets of substitutions (or "assignments") as "witnesses" occurs already in Van den Berg [**2, 4**], although the computational metaphor is left undeveloped there. We will return to Van den Berg [**2, 4**] later on, but for now let us briefly outline the present proposal.

Discourse markers arising from quantification are generated below by a deterministic program of the form

$$(*) \qquad (x :=? \ ; \ \pi)_\forall ,$$

where, roughly, the program π yields the *discourse representation structure* K in Kamp and Reyle [**14**]'s *abstraction* condition

$$X = \sum x : K .$$

The construct $\cdot_\forall$ has the additional virtue that it can be understood without mentioning substitutions or variables, and that, moreover, $(*)$ keeps the functional dependencies of x. Moving on, if a set of substitutions can grow by a program $(x :=?; \pi)_\forall$, that set can also be trimmed by a program

$$\mathcal{Q}x.\pi \ ,$$

where $\mathcal{Q}$ is some generalized quantifier. The program $\mathcal{Q}x.\pi$ (which will play a role similar to Kamp and Reyle [**14**]'s duplex condition) is essentially π, except that the conjunctive (universal) force of π on an input set U of substitutions is weakened so that not all substitutions in U need to execute π. Rather, we require that U be "$\mathcal{Q}x$-related" to *the* set $W \subseteq U$ of substitutions in U that can halt on π. The italics on "the" is meant to mark the "determinate"[1] character of the resulting discourse marker, although an "indeterminate" alternative will also be presented below. In either case, the condition $W \subseteq U$ builds in the conservativity of generalized quantifiers. Our interpretation of generalized quantifiers will be assembled from the programs $(x :=?; \pi)_\forall$ and $\mathcal{Q}x.\pi$, taking into account fine points concerning distributive-versus-collective and weak-versus-strong readings. Beyond these linguistic concerns, there are (inevitably?) various formal options that seem to carry no linguistic content, but which nevertheless demand our attention.

With these options as well as the question (Q2) of computability in mind, let us consider the passage (analyzed in Fernando [**9**]) from DPL transitions (defined relative to a fixed first-order model) to DRT transitions between discourse representation structures (DRS's). Observe that DRT transitions between DRS's not only

(S1) refer to transitions from various first-order models (in the same manner that first-order formulas apply to various first-order models),

but also

(S2) "determinize" the non-deterministic transitions in DPL (following the classic construction of deterministic finite automaton from non-deterministic finite automaton).

This suggests a two-step passage (spelled out in the concluding section below) from DPL transitions to DRT transitions that can drown out certain details of a DPL transition. The internalization in (S2) of the non-determinism of DPL transitions results in a somewhat more complex notion of state (corresponding to a DRS), built from "disjunctive" non-deterministic sets, as well as dual "conjunctive" parallel sets. Item (S1) points to a certain uniformity in DPL, upon which our answer to (Q2) will be founded. Our contention is that the notion of computation underlying DPL (and DRT) should not be construed in the *local* sense of ordinary (classical) recursion theory (a theory that applies

[1] The term "determinate" is intended to refer here to something akin to both defini*te* descriptions and *determin*istic programs. Its antonym "indeterminate" is meant to refer, in the same way, to "unselective" binding, as used in Van der Does [**6**] (assuming the present author understands Van der Does' intentions correctly).

specifically to *the* standard model of arithmetic; viz., the natural numbers), but in the *global* sense of Gödel's completeness theorem (as opposed to his incompleteness theorems). The terms "program" and "computation" are perhaps somewhat misleading when used in DPL; because the set Π of programs in DPL is closed under negation, there can be no program in Π universal for Π (contrary to the tradition of recursion theory). If, however, the term "program" is replaced by "formula" (and "computation" by "logic"), then this failure of universality is hardly surprising, in view of Tarski's theorem on the undefinability of truth. (Taking the natural numbers as a concrete example, negation leads to non-arithmeticity.) On the other hand, if one insists in using the words "program" and "computation", then — and this is our answer to (Q2) — DRT provides for DPL an appropriate *representational* level of abstraction for such discourse. Computation lives on syntax, and syntax is what DRT supplies in the form of a DRS. The additional computational power that a generalized quantifier $\mathcal{Q}$ might be expected to contribute should be understood at the global level, where uniformities in the map from a set M to the binary relation $\mathcal{Q}_M$ on Power(M) matter.

The remainder of the present paper is organized as follows. After some technical background on dynamic semantics in section 1, a bit of linguistic data is presented in section 2. Section 3 then considers the passage from S to Pow$_+(S)$, concentrating on propositional rather than quantificational matters. A closer look inside substitutions is needed in section 4 for the analysis (proper) of generalized quantifiers. Section 5 turns to some related work (viz., Van den Berg [**2, 4**] and Van der Does [**6**]), before section 6 concludes with a review of the essential points of the present work.

1. Between formulas and programs

Let Φ be a set of formulas φ, and Π be a set of programs π. Assume that the formulas and programs are interpreted relative to a fixed set S of states by functions $|\cdot|$ and ρ respectively such that for every $\varphi \in \Phi$, $|\varphi| \subseteq S$, while for every $\pi \in \Pi$, $\rho(\pi) \subseteq S \times S$, the intuition being that

$$|\varphi| \quad \approx \quad \{s \in S \mid \varphi \text{ is true at } s\}$$

and

$$\begin{aligned}\rho(\pi) \quad \approx \quad \{(s,t) \in S \times S \mid & \text{ given input state } s, \\ & \pi \textit{ can} \text{ halt on output state } t\}\ .\end{aligned}$$

Beyond the broad question of how formulas relate to programs, the present paper is concerned with what the word "can" (in the line above for $\rho(\pi)$) means in the presence of non-determinism.

Consider the embedding $\cdot ? : \Phi \to \Pi$ of formulas φ into programs (tests) $\varphi?$, given by $\rho(\varphi?) = \Delta|\varphi|$, where ΔU is the diagonal $\{(s,s) \mid s \in U\}$. Going the opposite direction, define the projection $\langle\cdot\rangle\top : \Pi \to \Phi$ of programs π into formulas $\langle\pi\rangle\top$, by $|\langle\pi\rangle\top| = dom\rho(\pi)$. A system of dynamic semantics such as

DPL amounts to a translation of formulas φ into programs π_φ, that in turn assert $\langle\pi_\varphi\rangle\top$. Now, certainly, for the case where S is the set of substitutions (mentioned in the introduction), the usual Tarskian definition can be obtained by setting

$$\begin{aligned} |\varphi\&\psi| &= |\langle\varphi?;\psi?\rangle\top| \\ |\sim\varphi| &= |\langle\neg(\varphi?)\rangle\top| \\ |\exists x\varphi| &= |\langle x:=?;\varphi?\rangle\top| \ , \end{aligned}$$

where

$$\begin{aligned} \rho(\pi;\pi') &= \rho(\pi)\circ\rho(\pi') \\ &(= \{(f,g)\in S\times S \mid (\exists h)\ f\ \rho(\pi)\ h \ \text{ and} \\ &\qquad\qquad h\ \rho(\pi')\ g\}) \\ \rho(\neg\pi) &= \{(f,f) \mid f\in S - dom\rho(\pi)\} \end{aligned}$$

and $\rho(x:=?)$ is as described in the introduction. It turns out, however, that the translation $\varphi\mapsto\varphi^{DPL}$ given by

$$\begin{aligned} R(\bar{t})^{DPL} &= R(\bar{t})? \\ (\varphi\&\psi)^{DPL} &= \varphi^{DPL}\ ;\ \psi^{DPL} \\ (\sim\varphi)^{DPL} &= \neg(\varphi^{DPL}) \\ (\exists x\varphi)^{DPL} &= x:=?\ ;\ \varphi^{DPL} \end{aligned}$$

has a different effect than the translation $\varphi\mapsto\varphi?$ in that the proposition $\langle\varphi^{DPL}\rangle\top$ (asserted by φ^{DPL}) is not equivalent (under ρ) to the proposition $\langle\varphi?\rangle\top$ (asserted by $\varphi?$). The reason, quite simply, is that although a formula φ is equivalent to $\langle\varphi?\rangle\top$, a program π may differ from $\langle\pi\rangle\top?$, the latter of which is equivalent to the static projection $\neg\neg\pi$ (interpreted as $\Delta dom\rho(\pi)$). The dynamic effect of $x:=?$ is useful for introducing a (singular) pronoun by an existential quantifier, as exemplified by the notorious donkey sentence

if a farmer owns a donkey then he beats it ,

which can be formalized naively as

$$(*) \qquad (\exists x,y)\ (\mathsf{f}(x)\ \&\ \mathsf{o}(x,y)\ \&\ \mathsf{d}(y)) \quad\supset\quad \mathsf{b}(x,y)\ .$$

DPL translates $(*)$, in turn, to the program

$$\neg\ (x:=?;y:=?;\mathsf{f}(x)?;\mathsf{o}(x,y)?;\mathsf{d}(y)?\ ;\ \neg\mathsf{b}(x,y)?)\ ,$$

following the equalities

$$\begin{aligned} (\varphi\supset\psi)^{DPL} &= (\sim(\varphi\&\sim\psi))^{DPL} \\ &= \neg(\varphi^{DPL}\ ;\ \neg\psi^{DPL})\ . \end{aligned}$$

The program scheme $\neg(\pi;\neg\pi')$ generated by the DPL translation of implication occurs again in the DPL translation of universal quantification

$$\begin{aligned} (\forall x\varphi)^{DPL} &= (\sim\exists x\sim\varphi)^{DPL} \\ &= \neg(x:=?\ ;\ \neg\varphi^{DPL})\ . \end{aligned}$$

The outer negation blocks the introduction of discourse markers (for plurals).[2] To overcome the static effect of negation $\neg$, a binary program construct $\Rightarrow$ is introduced in Fernando [**10**] that externalizes the dynamism within $\neg(\pi;\neg\pi')$ by collecting (in the output) witnesses g' to the $\forall\exists$-clause in

$$\begin{array}{lll} f\ \rho(\neg(\pi;\neg\pi'))\ g & \text{iff} & f = g \text{ and } (\forall f' \text{ s.t. } f\ \rho(\pi)\ f') \\ & & \qquad (\exists g')\ f'\ \rho(\pi')\ g'\ , \end{array}$$

the idea being that the "process f spawns" the processes g'. The first proposal in Fernando [**10**] is

(P1) to introduce functional states:

$$\begin{array}{lll} s\ [\![\pi \Rightarrow \pi']\!]_1\ F & \text{“iff”} & (s \notin \mathit{dom}[\![\pi]\!]_1 \text{ and } s = F) \text{ or} \\ & & (s \in \mathit{dom}[\![\pi]\!]_1 \text{ and } F \text{ is a} \\ & & \ \text{function with domain} \\ & & \ \{s' \mid s[\![\pi]\!]_1 s'\} \text{ s.t.} \\ & & \ (\forall s' \text{ s.t. } s[\![\pi]\!]_1 s') \\ & & \qquad s'[\![\pi']\!]_1 F(s'))\ . \end{array}$$

Implicit above is a passage from S to the least solution S_1 to $S_1 = S \cup (S_1 \rightharpoonup S_1)$, where $A \rightharpoonup B$ is the set of partial functions from A to B. A relation $R \subseteq S \times S$ can be lifted canonically to $\hat{R}$ by requiring that for functions $F \in S_1 - S$ (and for all $G \in S_1$),

$$\begin{array}{lll} F\ \hat{R}\ G & \text{iff} & \mathit{dom}(F) = \mathit{dom}(G) \text{ and} \\ & & (\forall s \in \mathit{dom}(F))\ F(s)\ \hat{R}\ G(s)\ . \end{array}$$

A simplification of (P1) is

(P1.5) to replace the function F by its image U

$$\begin{array}{lll} s\ [\![\pi \Rightarrow \pi']\!]_{1.5}\ U & \text{“iff”} & (s \notin \mathit{dom}[\![\pi]\!]_{1.5} \text{ and } s = U) \\ & & \text{or } (s \in \mathit{dom}[\![\pi]\!]_{1.5} \\ & & \ \text{and there is a function} \\ & & \ F : \{s' | s[\![\pi]\!]_{1.5} s'\} \rightarrow_{onto} U \\ & & \ \text{s.t. } (\forall s' \text{ s.t. } s[\![\pi]\!]_{1.5} s') \\ & & \qquad s'[\![\pi']\!]_{1.5} F(s'))\ . \end{array}$$

That is, pass from S to the least solution $S_{1.5}$ to $S_{1.5} = S \cup \mathrm{Pow}_+(S_{1.5})$. A further simiplification of (P1) is

[2] The reader familiar with dynamic logic can cast $(\forall x\varphi)^{DPL}$ as $[x :=?]\langle\varphi^{DPL}\rangle\top?$, and view the present problem as identifying a program counterpart $\Rightarrow$ to the box/square modalities, playing a role analogous to that of ; for the diamond/angle modalities

$$\begin{array}{lll} (\langle\pi\rangle\varphi)? & \equiv & \neg\neg(\pi\ ;\ \varphi?) \\ ([\pi]\varphi)? & \equiv & \neg\neg(\pi \Rightarrow \varphi?)\ , \end{array}$$

where $\equiv$ is semantic equivalence.

(P2) to project $S_{1.5}$ into $\text{Pow}_+(S)$ by embedding S into $\text{Pow}_+(S)$ via $f \mapsto \{f\}$ and collapsing $\text{Pow}_+(\text{Pow}_+(S))$, $\text{Pow}_+(\text{Pow}_+ (\text{Pow}_+(S)))$, etc. to $\text{Pow}_+(S)$ by suitable iterations of union. That is, project $S_{1.5}$ into $\text{Pow}_+(S)$ by the map $\tilde{\cdot}$ defined by

$$\tilde{s} = \begin{cases} \{s\} & \text{if } s \in S \\ s & \text{if } s \in \text{Pow}_+(S) \\ \bigcup\{\tilde{t} \mid t \in s\} & \text{otherwise.} \end{cases}$$

There are at least two complications facing the proposals above.

(i) First, the map $R \mapsto \hat{R}$ in (P1) does *not* commute with, for example, the interpretation $\cup$ of $+$,

$$(R \hat{\cup} R') \quad \text{need not equal} \quad \hat{R} \cup \hat{R}' \ .$$

Neither does it commute with $\neg$. Nor do matters improve when passing to (P1.5) or to (P2).

(ii) In (P1.5) or (P2), it is not clear how, proceeding from (P1), to collapse S_1 to $S_{1.5}$ or S_2, without the information (in the mapping F) about how processes are spawned. In place of the canonical map $R \mapsto \hat{R}$ of (P1), there are (as explained in section 3) at least two non-equivalent ways of lifting a binary relation on S to a binary relation on $\text{Pow}_+(S)$, consistent with a conjunctive reading of the non-empty sets of substitutions.

As will be shown below, item (i) carries genuine linguistic content (pointing, as it does, to the difference between so-called collective and distributive readings). Whether or not item (ii) does is unclear to the present author.

2. Some linguistic data

Sentences such as the following suggest that singular pronouns tend to have indeterminate references, while plural pronous introduced by generalized quantifiers tend to be determinate.

(1) A/some congressman admires Kennedy. He is very junior.

(2) Some congressmen admire Kennedy. They are very junior.

"He" in (1) is some (indeterminate) congressman who admires Kennedy. "They" in (2) is, according to Evans [**8**], an E-type pronoun referring to all congressmen who admire Kennedy. Its determinateness suggests that a plural object must be definable (possibly in first-order logic). The truth conditions for (2) contrast with those for

(3) Some congressmen admire Kennedy and are very junior.

(3) has the same truth conditions as

(4) Some congressmen admire Kennedy. Some of them are very junior.

However, "they" would seem to have different available meanings in (3) and (4): follow up (4) with

(4b) Most are quite senior.

A somewhat minor point is whether a discourse marker should be kept.

(5) Some congressmen admire Kennedy. Two of them are very junior. Three are somewhat senior. Both are in their early thirties.

(4b) and (5) suggest that discourse markers should not be destroyed (but perhaps copied to some new variable).

In fact, the determinatenes of generalized quantifiers is somewhat controversial. It is felt by some that the line

two congressmen admire Kennedy

does *not* imply that at most two congressmen admire Kennedy. (Consider the analysis in Kamp and Reyle [**14**], p. 378 of "two of them" by some indeterminate V.) But then an E-type reading of

Two congressmen admire Kennedy. They are very junior.

makes "they" refer to all congressmen who admire Kennedy. Even if we concede that "two" is ambiguous between "exactly two" and "at least two", the creation of a discourse marker "they" suggests that there are exactly two such congressmen. Or is an indeterminate choice of such a pair from a model where there are more than two allowed? Both options will be explored below.

Then there is also the question of what it means for the group to admire Kennedy. Even if, simplifying matters, we agree that each one in the group must admire Kennedy, there is more trouble ahead. The sentence

(6) they are very junior or very silly

has the collective reading

all of them are very junior, or all of them are very silly

(where "or" has scope outside "they"), and the distributive reading

each one of them is either very junior or very silly

(where "or" has scope inside "they"). Similarly, turning to negation,

(7) they are not very junior

has the distributive reading

each one of them is not very junior ,

and the collective reading

at least one of them is not very junior .

There will be no avoiding such complications in our formalization.

Finally, let us turn to a point often made about the proposal in Fernando [**10**], on which the present paper builds. Although it is true enough that the line

Each student told a joke. She forgot the punchline.

(from Barwise [**1**]) sounds anomalous, suggesting that the discourse marker for a noun phrase that is universally quantified should not be available beyond the sentence in which it occurs, it is equally true that lines such as

Every player chooses a pawn. He puts it on square one.

(from Groenendijk and Stokhof [**11**]) suggest otherwise. The former suggestion is available in Fernando [**10**] via the usual classical reduction (adopted in DPL) of $\forall$ and $\supset$ in terms of negation, $\exists$ and conjunction. The latter suggestion is also within the scope of Fernando [**10**]. But alas, just how to choose between these two options is not. The present paper will be full of this sort of embarrassment of riches.

3. Second-order programs

The present section is concerned with a passage from relations on S to relations on $\mathrm{Pow}_+(S)$, pictured by

$$\frac{\rho(\pi) \subseteq S \times S}{[\![\pi]\!] \subseteq \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S)}$$

along the embedding $f \mapsto \{f\}$ of S into $\mathrm{Pow}_+(S)$, so that for the programs π from ordinary dynamic logic,

$$\{f\}[\![\pi]\!]\{g\} \quad \text{iff} \quad f\ \rho(\pi)\ g\ .$$

(Recall that the empty set $\emptyset$ is excluded as it arises from the failure of a substitution to make a ρ-transition.) There is some freedom in choosing exactly how to pass from ρ to $[\![\cdot]\!]$, even if we require that a set $U \in \mathrm{Pow}_+(S)$ of substitutions should be treated in a conjunctive (parallel) manner, dual to the disjunctive manner of non-determinism. A binary relation R on S can be lifted to a binary relation on $\mathrm{Pow}_+(S)$ as

$$\begin{aligned} \uparrow R \quad = \quad & \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \exists \text{ function } F \text{ from} \\ & \qquad U \text{ onto } V \text{ s.t. } (\forall s \in U)\ s\ R\ F(s)\} \end{aligned}$$

(which is implicit in the interpretation of ; in Peleg [**16**]) or else (non-equivalently) as

$$\begin{aligned} \Uparrow R \quad = \quad & \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \\ & (\forall s \in U)(\exists t \in V)\ sRt \text{ and } (\forall t \in V)(\exists s \in U)\ sRt\}\ . \end{aligned}$$

Comparing these with the the inductive clauses from dynamic logic, note that $(\uparrow R) \circ (\uparrow R')$ need not equal $\uparrow (R \circ R')$, as pointed out to the author by A. Visser. On the other hand, $(\Uparrow R) \circ (\Uparrow R')$ is $\Uparrow (R \circ R')$, at the cost of some looseness in the embedding $f \mapsto \{f\}$ (already for random assignments).

Unfortunately, neither $\uparrow$ nor $\Uparrow$ commutes with $\cup$, although it is easy enough to motivate the failure here by the ambiguity in sentence (6) from the previous section. Similar complications surround negation, as illustrated by (7).

For the sake of concreteness, let us agree that if π is a random assignment $x :=?$ or a test $\varphi?$, then $[\![\pi]\!] = \uparrow \rho(\pi)$. Then, proceeding inductively, define (for collective readings)

$$\begin{aligned} [\![\pi;\pi']\!] &= [\![\pi]\!] \circ [\![\pi']\!] \\ [\![\pi+\pi']\!] &= [\![\pi]\!] \cup [\![\pi']\!] \\ [\![\neg\pi]\!] &= \{(s,s) \mid s \in \mathrm{Pow}_+(S) - dom[\![\pi]\!]\} \ , \end{aligned}$$

further closing the programs (if the reader desires) under unary program constructs $\cdot_\uparrow$ and $\cdot_\Uparrow$, subject to

$$\begin{aligned} [\![\pi_\uparrow]\!] &= \{(U,V) \mid U \in \mathrm{Pow}_+(S) \text{ and there is a} \\ &\qquad \text{function } F \text{ with domain } U \text{ s.t. } V = \bigcup_{s \in U} F(s) \\ &\qquad \text{and } (\forall s \in U)\ \{s\}[\![\pi]\!]F(s)\} \\ [\![\pi_\Uparrow]\!] &= \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid (\forall s \in U) \\ &\qquad (\exists W \subseteq V)\ \{s\}[\![\pi]\!]W \text{ and} \\ &\qquad (\forall t \in V)(\exists W \subseteq V)(\exists s \in U) \\ &\qquad \{s\}[\![\pi]\!]W \text{ and } t \in W\} \ . \end{aligned}$$

This allows a formulation of distributive alternatives $(\pi;\pi')_\uparrow$, $(\pi+\pi')_\uparrow$, $(\pi+\pi')_\Uparrow$, and $(\neg\pi)_\uparrow$ $(= (\neg\pi)_\Uparrow)$ to $\pi;\pi'$, $\pi+\pi'$, etc. Turning finally to the reason for passing from S to $\mathrm{Pow}_+(S)$, define

$$\begin{aligned} U[\![\pi \Rightarrow \pi']\!]V \quad \text{iff} \quad & (U \notin dom[\![\pi]\!] \text{ and } U = V) \text{ or} \\ & (U \in dom[\![\pi]\!] \text{ and there is a function } F \\ & \text{ with domain } \{U' \mid U[\![\pi]\!]U'\} \text{ s.t.} \\ & \quad V = \bigcup\{F(U') \mid U' \in dom(F)\} \text{ and} \\ & \quad (\forall U' \in dom(F))\ U'[\![\pi']\!]F(U')) \ . \end{aligned}$$

Observe that

$$\begin{aligned} [\![\neg\neg(\pi \Rightarrow \pi')]\!] &= [\![\neg(\pi;\neg\pi')]\!] \\ [\![\neg\pi]\!] &= [\![\pi \Rightarrow \bot]\!] \end{aligned}$$

(where $[\![\bot]\!] = \emptyset$), although the equivalence between $\neg\pi$ and $([\pi]\bot)?$ is lost. A program counterpart $\pi_\forall$ to the box modality $[\pi]$ on formulas of dynamic logic that turns the usual "or" branching of Kripke frames into "and" branching,

$$U[\![\pi_\forall]\!]V \quad \text{iff} \quad V = \bigcup\{U' \mid U[\![\pi]\!]U'\} \ (\neq \emptyset) \ ,$$

can then be derived as

$$\pi_\forall \quad = \quad \neg\neg\pi \ ; \ (\pi \Rightarrow \top) \ ,$$

where $\top = \neg\bot$. Going the opposite direction,

$$[\![\pi \Rightarrow \pi']\!] \quad = \quad [\![(\pi_\forall ; \pi') + \neg\pi]\!]$$

provided π' is distributive in the sense of $\uparrow$ (i.e., $[\![\pi']\!] = [\![\pi'_\uparrow]\!]$) or π is deterministic. Although random assignments and tests are defined to be distributive, it is easy to manufacture a non-deterministic choice $\pi_1 + \pi_2$ (e.g., from tests) that is not.

Of the two ways of turning an arbitrary program into a deterministic program, $\neg\neg\cdot$ is static, whereas $\cdot_\forall$ is dynamic. Notice that if π is deterministic, then $[\![\pi]\!] = [\![\pi_\forall]\!]$. In particular, as $\neg\pi$ is always deterministic, whether or not π is, $[\![\neg\pi]\!] = [\![(\neg\pi)_\forall]\!]$. It is also easy to see that $[\![\neg\pi]\!] = [\![\neg(\pi_\forall)]\!]$. Recalling the construct $\cap$ of Peleg [**16**], note that if π and π' are deterministic, then $\pi \cap \pi'$ can be coded as $(\pi + \pi')_\forall$.

4. Generalized quantifiers

Given a generalized quantifier $\mathcal{Q}$ that maps the universe M of a first-order model to a binary relation $\mathcal{Q}_M$ on Power(M), Chierchia [**5**] (as well as Van Eijck and De Vries [**7**]) extend quantified dynamic logic by binary programs $\mathcal{Q}x(\pi, \pi')$ (for $x \in X$ and programs π and π'), interpreting it (by an extension of ρ) as the externally static test

$$\begin{aligned}\{(s,s) \quad | \quad & s \in S \text{ and } \{m \in M \mid \exists s' \ s[x/m] \ \rho(\pi) \ s'\} \ \mathcal{Q}_M \\ & \{m \in M \mid \exists s' \ s[x/m] \ \rho(\pi;\pi') \ s'\}\} \ .\end{aligned}$$

This corresponds to what Chierchia [**5**] calls the "weak" reading, which is presented as line (4.166) in p. 421 of Kamp and Reyle [**14**], for the semantics of duplex conditions. The alternative "strong" reading

$$\begin{aligned}\{(s,s) \quad | \quad & s \in S \text{ and } \{m \in M \mid \exists s' \ s[x/m] \ \rho(\pi) \ s'\} \ \mathcal{Q}_M \\ & \{m \in M \mid \exists s' \ s[x/m] \ \rho(\neg(\pi;\neg\pi')) \ s'\}\} \ .\end{aligned}$$

(corresponding to line (4.172), p. 424 of Kamp and Reyle [**14**]) is also offered in Chierchia [**5**] (and Van Eijck and De Vries [**7**]). In either case, a basic point of the present work is to externalize the dynamic effects so as to support the imposition of further constraints on variables introduced by generalized quantifiers. That is, our analysis of a generalized quantifier $\mathcal{Q}$ will incorporate the abstraction condition of Kamp and Reyle within our mechanism for their duplex condition. We will modify the notation $\mathcal{Q}x(\pi, \pi')$ slightly by writing $(\mathcal{Q}x : \pi)\pi'$ for our proposal.

Concentrating first on the strong reading, consider the interpretation

$$(All\ x : \pi)\pi' \quad = \quad (x := ?; \pi) \ \Rightarrow \ \pi'$$

implicit in Fernando [**10**]. This is semantically equivalent, if $[\![\pi']\!] = [\![\pi'_\uparrow]\!]$, to

$$(x := ?; \pi)_\forall \ ; \ \pi' \quad + \quad \neg(x := ?; \pi) \ .$$

For the remainder of this section, let us not be distracted by the question of the distributivity of π', as this can always be arranged by replacing π' by

$\pi'_{\uparrow}$. (Similarly, throughout this section, the non-determinism suggested by $+$ will be spurious, as the disjuncts will be mutually exclusive.) Using certain programs that will be explained shortly, let us generalize the interpretation above of *All* to $\mathcal{Q}$ by equating $(\mathcal{Q}x : \pi)\pi'$ with

$$\begin{array}{l} (x := ?; \pi)_{\forall} \; ; \; (\mathcal{Q}x.\pi' \; + \; \langle \mathcal{Q}^{-}x\rangle; \neg\pi') \qquad + \\ \qquad\qquad\qquad \neg(x := ?; \pi) \; ; \; \langle \mathcal{Q}^{0}x\rangle \end{array}$$

where $All\ x.\pi$ is (under $[\![\cdot]\!]$) just π, $\langle All^{-}x\rangle$ is $\bot$ (i.e., $[\![\langle All^{-}x\rangle]\!]$ is $\emptyset$), and $\langle All^{0}x\rangle$ is $\top$ (i.e., $[\![\langle All^{0}x\rangle]\!]$ is equality on $\mathrm{Pow}_{+}(S)$). More generally, the new programs $\mathcal{Q}x.\pi$ (for every program π), $\langle \mathcal{Q}^{-}x\rangle$, and $\langle \mathcal{Q}^{0}x\rangle$ are interpreted on the basis of the relation

$$\begin{array}{rcl} \mathcal{Q}_M x & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \;\mid \\ & & \quad \{s(x) \mid s \in U\} \; \mathcal{Q}_M \; \{s(x) \mid s \in V\}\} \end{array}$$

as follows

$$\begin{array}{rcl} [\![\mathcal{Q}x.\pi]\!] & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \mid \\ & & \quad U \; \mathcal{Q}_M x \; \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \;\; \text{and} \\ & & \quad \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \; [\![\pi]\!] \; V\} \\ [\![\langle \mathcal{Q}^{-}x\rangle]\!] & = & \{(U,U) \mid U \in \mathrm{Pow}_{+}(M) \text{ and } U \; \mathcal{Q}_M x \; \emptyset\} \\ [\![\langle \mathcal{Q}^{0}x\rangle]\!] & = & \{(U,U) \mid U \in \mathrm{Pow}_{+}(M) \text{ and } \emptyset \; \mathcal{Q}_M x \; \emptyset\} \; . \end{array}$$

Conservativity is essentially built in, not only in the semantic clause for $\mathcal{Q}x.\pi$ (via the condition $W \subseteq U$), but also in the test $\langle \mathcal{Q}^{0}x\rangle$, which is an all-or nothing affair (being either $\top$ or $\bot$). The intuition behind $\mathcal{Q}x.\pi$ is that π is executed without necessarily insisting that every substitution in the input state U can halt on π, but rather that U is $\mathcal{Q}_M x$-related to the set of substitutions in U that can halt on π. The program $\mathcal{Q}x.\pi$ can be decomposed as the sequential composition

$$\mathcal{Q}x.\neg\neg\pi \quad ; \quad \pi$$

of a deterministic program $\mathcal{Q}x.\neg\neg\pi$ with π, since $[\![\mathcal{Q}x.\neg\neg\pi]\!]$ incorporates the first of $[\![\mathcal{Q}x.\pi]\!]$'s two conjuncts, plus a tiny bit more

$$\begin{array}{rcl} [\![\mathcal{Q}x.\neg\neg\pi]\!] & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \;\mid \\ & & \quad V = \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \;\; \text{and} \\ & & \quad U \; \mathcal{Q}_M x \; V \;\; \text{and} \;\; V \in dom[\![\pi]\!]\} \; . \end{array}$$

Although the unary program constructs $\mathcal{Q}x.\cdot$ and $\mathcal{Q}x.\neg\neg\cdot$ are, in the presence of sequential composition and negation, interdefinable (so that either could be taken as primitive), it is quite another matter (as we will see shortly) to reduce these unary constructs to a nullary construct $\langle \mathcal{Q}^{+}x\rangle$, independent of π.

The external dynamic effects of a program fragment π can always be suppressed by prefixing the appropriate program fragment by double negation $\neg\neg$.

Such effects are, of course, crucial in interpreting inter-sentential anaphora, which allows constraints on pronouns to be imposed by incremental updates. Double negation $\neg\neg$ is useful for encoding the weak reading of $(\mathcal{Q}x : \pi)\pi'$ as

$$(x :=?; \neg\neg\pi)_\forall \ ; \ (\mathcal{Q}x.(\pi;\pi') \ + \ \langle \mathcal{Q}^- x\rangle; \neg(\pi;\pi')) \quad + \\ \neg(x :=?; \pi) \ ; \ \langle \mathcal{Q}^0 x\rangle \quad .$$

Because the dynamic effects of π are suppressed in $(x :=?; \neg\neg\pi)_\forall$, the program π must (in contrast to the strong reading) be repeated in $(\mathcal{Q}x.(\pi;\pi') + \langle \mathcal{Q}^- x\rangle; \neg(\pi;\pi'))$.

Next, let us turn to a deviant treatment of quantification, resulting from replacing $\mathcal{Q}x.\pi$ by the sequential composition

$$\langle \mathcal{Q}^+ x\rangle \ ; \ \pi$$

of a program $\langle \mathcal{Q}^+ x\rangle$ with π, where the program $\langle \mathcal{Q}^+ x\rangle$ chooses *some* non-empty subset V of the input set U to which U is $\mathcal{Q}_M x$-related

$$[\![\langle \mathcal{Q}^+ x\rangle]\!] \quad = \quad \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \\ U \mathcal{Q}_M x V \text{ and } V \subseteq U\}.$$

The sequential composition with π then means that V must execute π.[3] Setting aside dynamic complications, to say that for all sets A and B,

$$(*) \qquad A\mathcal{Q}_M B \quad \text{iff} \quad (\exists C) \ \ A\mathcal{Q}_M C \ \text{ and } \ C \subseteq B$$

is to suppose that $\mathcal{Q}$ is upwards-monotone on its second argument (e.g. $\forall, \exists$). Even in that case, however, the discourse marker ("they") introduced by the quantification must refer (according to the E-type account) to $A \cap B$, and not to $A \cap C$, for some indeterminate choice of C. If $\mathcal{Q}$ is not upwards-monotone on its second argument, then $(*)$ fails even under a modification of $(*)$ where C is required to be maximal (as can be seen by taking, for example, $\mathcal{Q}$ to be "exactly two"). If the non-deterministic program $\langle \mathcal{Q}^+ x\rangle$ is ruled out, then it is plausible that the only sets of substitutions required (as inputs to and outputs of programs) are first-order definable, with parameters (for the "singular" variables).

5. Related work

As has already been pointed out in the introduction, the present work borrows from Peleg [**16**] the idea of a "conjunctive" set, but differs from it by introducing such sets not by $\cap$ but by $\Rightarrow$ (to capture a notion of parallelism dual to the non-determinism in $x :=?$, rather than in $+$). Conjunctive sets also appear in Van den Berg [**2, 4**], but again not via $\Rightarrow$ (or $\cdot_\forall$). Instead, a 3-valued logic is employed in Van den Berg [**2, 4**] to make a "plural logic" dynamic (although the present author cannot claim to understand exactly how this feat is carried out).

[3] A comparison with the indeterminate treatment of "two of them" in Kamp and Reyle [**14**], p. 378 is instructive here.

Next, let us say a few words about the system *Dynamic Quantifier Logic* (DQL) of Van der Does [**6**]. A basic premise of DQL is that singular pronouns involve "unselective binding"[4], while plurals are given by (E-type) descriptions (rather than some possibly indeterminate notion of maximality). Given a model with universe D, DQL interprets formulas relative to triples $(\mathbf{a}, \mathbf{c}, \mathbf{h})$ of

(i) *assignments* **a**; i.e., functions from the set of variables to D (called substitutions above);
(ii) *contexts* **c**; i.e., partial functions from the set of variables to formulas;

and

(iii) *choice functions* **h**; i.e., functions from subsets of D to D (used for unselective binding on singulars).

(Assignments **a** are merged with choice functions **h** in Van der Does' later work, although this is a rather minor point.) The main idea, as Van der Does writes in his abstract for the Ninth Amsterdam Colloquium, is as follows.

> In my system, as in others, pronouns are quantifiers which depend for their interpretation on parametrized context sets. Plural pronouns are interpreted as a universal quantifier over such a context set, while singular pronouns involve a choice from such a set ...

Is he thus committed to a distributive reading of plurals? Whether or not he is, the general picture seems to be in accord with our determinate analysis above. A somewhat alarming difference, however, is that DQL "is best thought as a standard 'static' logic, which comes with a separate dynamic module to handle the context generated by a text" (Van der Does [**6**], p. 20); "DQL uncouples the interpretation of a formula from its context change potential" (p.25). The context change potentials associated with formulas are functions from contexts to contexts. If, however, transitions are given by *functional* context change potentials, what becomes of the non-determinism in DPL transitions? As the non-determinism in DPL arises (only) from the random assignments used for singular pronouns, presumably DQL has no trouble dealing with that via its choice functions. But alas, the present author must confess that it is not clear to him (at the time of writing) exactly how DQL relates to DPL, let alone to the extension of DPL proposed above.

6. Discussion

The extension to DPL proposed above is based on

(i) generating "conjunctive" sets U of substitutions through a program construct $\cdot_\forall$,

and

(ii) weakening the "universal" force of a program π on such a set U according to a generalized quantifier $\mathcal{Q}$ by forming the program $\mathcal{Q}x.\pi$.

[4] Roughly, indeterminateness; see footnote 1.

The reader familiar with chapter 4 of Kamp and Reyle [**14**] might regard $\cdot_\forall$ as a mechanism for abstraction, and $\mathcal{Q}x.\cdot$ as a tool for the duplex condition.[5] But, in fact, the constructs $\cdot_\forall$ and $\mathcal{Q}x.\cdot$ can be explained without mentioning DRT — and, in the case of $\cdot_\forall$, without mentioning variables —, proceeding instead from computational intuitions involving a form of duality between non-determinism and parallelism. The linguistic relevance of the present work rests on the degree to which the distinction between definite and indefinite noun phrases can be captured by that between deterministic and non-deterministic programs. By providing for conjunctive branching, $\cdot_\forall$ yields, from a dynamic viewpoint, more interesting deterministic programs than the static projection $\neg\neg\cdot$. Furthermore, it is a trivial matter to derive $\cdot_\forall$ from a binary program construct $\Rightarrow$ introduced in Fernando [**10**] to extend the DPL translation.

As earnestly as the present author has tried to motivate his proposals, the reader sensitive to "ad hoc-kery" might worry all the same about some five branch-points that were faced along the way:

(B1) the choice in section 1 (of an extension to higher types) between proposals (P1), (P1.5) and (P2)
(B2) within (P1.5) and (P2), the choice (of a lifting from relations on S to relations on $\mathrm{Pow}_+(S)$) between $\uparrow\cdot$ and $\Uparrow\cdot$ (in section 3)
(B3) distributive versus collective readings
(B4) weak versus strong readings

and

(B5) the determinateness or indeterminateness of discourse markers introduced by generalized quantifiers.

Of (B1) to (B5), the last three obviously have, for a linguist, some substance. As for the first two points, however, the question is *who cares*? Whether these options matter can be investigated (by formulating items (S1) and (S2) in the introduction) using the somewhat simple-minded notion of a *(pointed)*

[5] More precisely, the present author claims that if $\mathcal{K}$ is the set of DRS's given up to chapter 4 of Kamp and Reyle [**14**], and if Π is the set of programs from above, then a mapping $K \mapsto \pi_K$ can be defined from $\mathcal{K}$ to Π, with the property that

$$\pi_K \text{ sends the empty DRS to the DRS } K.$$

Briefly, the main point of the inductive definition of the mapping would be to encode

(i) abstraction $X = \sum x :\ K$ by $(x :=?; \pi_K)_\forall \ + \ \neg(x :=?\ ;\ \pi_K)$,

and

(ii) the duplex condition with restrictor K, quantifier $\mathcal{Q}$, principal discourse referent x and scope K' by $\neg\neg(\mathcal{Q}x : \pi_K)\pi_{K'}$, for some binary program construct $(\mathcal{Q}x : \cdot)\cdot$ defined as in section 4 using $\cdot_\forall$ and $\mathcal{Q}x.\cdot$.

As indicated in the introduction, $(x :=?; \pi_K)_\forall$ preserves the functional dependencies of x in K. Furthermore, inasmuch as $(\mathcal{Q}x : \cdot)\cdot$ already incorporates the effect of abstraction, it can be argued that $(\mathcal{Q}x : \cdot)\cdot$ is *not* subject to the criticism made in Van den Berg [**3**] "that the dynamic behavior of the quantifier is totally **ad hoc**. The definition gives a condition in terms of the generalized quantifier and on top of that it gives some condition on the output ..." (On the other hand, this very decomposition points to the conjecture that all sets of substitutions needed are first-order definable.)

transition system, a 4-tuple $\langle S, L, \rightarrow, s_0 \rangle$, where S and L are non-empty sets, whose elements are called *states* and *labels* respectively, $\rightarrow \subseteq S \times L \times S$ is a *transition predicate*, and $s_0 \in S$ is an *initial state*. The semantics of DPL can be characterized, relative to a first-order model $\mathbf{M}$ and an infinite set X of variables, by a transition system $[\![\mathbf{M}]\!] = \langle X \rightarrow_{fin} M, \Phi, [\![\cdot]\!]_{\mathbf{M}}, \emptyset \rangle$, where

(i) the state set $X \rightarrow_{fin} M$ is the set of $\mathbf{M}$-substitutions; i.e., functions from finite subsets of X to the universe M of $\mathbf{M}$,
(ii) the label set Φ is the set of first-order formulas (with the signature of $\mathbf{M}$) over the set X of variables,
(iii) the transition predicate $[\![\cdot]\!]_{\mathbf{M}}$ associates with every $\varphi \in \Phi$ the binary relation $[\![\varphi]\!]_{\mathbf{M}}$ on $X \rightarrow_{fin} M$, given by $\rho(\varphi^{DPL})$ (modified in the obvious way for finite functions), and
(iv) the initial state $\emptyset$ is the totally undefined function.

Similarly, the various options in (B1) and (B2) yield transition systems, albeit somewhat more complicated ones. A transition system for Kamp [**15**] can also be constructed, independent of some first-order model $\mathbf{M}$, by taking DRS's as its states, among which the empty DRS is initial; call that transition system $\mathcal{DRT}$. Notice that whereas transitions in $\mathcal{DRT}$ are deterministic, those in $[\![\mathbf{M}]\!]$ are not. But it is easy enough to make the transitions in $[\![\mathbf{M}]\!]$ deterministic by passing to the transition system $\mathcal{D}([\![\mathbf{M}]\!])$, given by the map $\mathcal{D}$ on transition systems that sweeps the non-determinism underneath states (formed as "disjunctive" sets, dual to the conjunctive sets above). (As mentioned in item (S2) of the introduction, $\mathcal{D}$ is essentially the classic construction of deterministic finite automata from non-deterministic ones.) It turns out (as shown in Fernando [**9**]) that the transition system $\mathcal{DRT}$ is isomorphic to the obvious transition system sum

$$\sum \{\mathcal{D}([\![\mathbf{M}]\!]) \mid \mathbf{M} \in \mathcal{M}\}$$

over the class $\mathcal{M}$ of (say countable) first-order models $\mathbf{M}$ (of the appropriate signature). (The formation of the sum above captures the uniformity mentioned in item (S1) of the introduction.) That is, as states go, DRS's embody a good deal of partiality and complexity, hiding ghastly details from the transition predicate. The transformations $\mathcal{D}$ and $\sum$ together bridge the gap between DPL and DRT (mentioned as items (S1) and (S2) in the introduction) that can be depicted as follows.

	DPL	**DRT**
states	substitution	DRS
transitions	non-deterministic	deterministic
	semantic	syntactic
	local	global
labels	formula	program

Now returning to the ad hoc-kery in (B1) and (B2) above, the point is to compare the transition systems they induce modulo $\mathcal{D}$ and $\sum$. Much work remains to be done.

Acknowledgments

My thanks to colleagues in Amsterdam (Martin van den Berg, Jan van Eijck), Utrecht (Kees Vermeulen, Albert Visser), Stuttgart (Ede Zimmermann) and Edinburgh (David Milward, Massimo Poesio) for comments on oral and/or written presentations of this work.

References

[1] Jon Barwise. Noun phrases, generalized quantifiers and anaphora. In E. Engdahl and P. Gärdenfors, editors, *Generalized Quantifiers*, Studies in Language and Philosophy. Dordrecht: Reidel, 1987.

[2] Martin van den Berg. Dynamic generalized quantifiers. In J.M. van der Does and J. van Eijck, editors, *Generalized Quantifier Theory and Applications.* Dutch Network for Language, Logic and Information Prepublication, Amsterdam, 1991. Revised version to appear as CSLI Lecture Notes.

[3] Martin van den Berg. A direct definition of generalized dynamic quantifiers. Preliminary abstract submitted to the Ninth Amsterdam Colloquium, 1993.

[4] Martin van den Berg. Full dynamic plural logic. In K. Bimbó and A. Máté, editors, *Proc. 4th Symposium on Logic and Language.* Budapest, 1993.

[5] Gennaro Chierchia. Anaphora and dynamic logic. ITLI Prepublication, University of Amsterdam, 1990.

[6] Jaap M. van der Does. The dynamics of sophisticated laziness. Manuscript, ILLC, Amsterdam, September 1993.

[7] J. van Eijck and F.J. de Vries. Dynamic interpretation and Hoare deduction. *J. Logic, Language and Information*, 1, 1992.

[8] Gareth Evans. Pronouns. *Linguistic Inquiry*, 11(2), 1980.

[9] Tim Fernando. Transition systems and dynamic semantics. In D. Pearce and G. Wagner, editors, *Logics in AI*, LNCS 633 (subseries LNAI). Springer-Verlag, Berlin, 1992. A slightly corrected version has appeared as CWI Report CS-R9217, June 1992.

[10] Tim Fernando. The donkey strikes back. In *Proc. of the 6th Conference of the European Chapter of the Association for Computational Linguistics.* OTS, Utrecht, The Netherlands, 1993.

[11] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[12] David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. D. Reidel, 1984.

[13] Irene Heim. The semantics of definite and indefinite noun phrases. Dissertation, University of Massachusetts, Amherst, 1982.

[14] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers (Dordrecht), 1993.

[15] Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language.* Mathematical Centre Tracts 135, Amsterdam, 1981.

[16] David Peleg. Concurrent dynamic logic. *J. Assoc. Computing Machinery*, 34(2), 1987.